

# The Effect of Scratch- and Lego Mindstorms Ev3-Based Programming Activities on Academic Achievement, Problem-Solving Skills and Logical-Mathematical Thinking Skills of Students

[1] Amasya University, Faculty of Technology, Turkey  
ozgen.korkmaz@amasya.edu.tr

Özgen KORKMAZ [1]

## ABSTRACT

The aim of this study was to investigate the effect of the Scratch and Lego Mindstorms Ev3 programming activities on academic achievement with respect to computer programming, and on the problem-solving and logical-mathematical thinking skills of students. This study was a semi-experimental, pretest-posttest study with two experimental groups and a control groups. A total of 75 students were divided among all three groups. Scratch-based game activities were applied to Study Group I. Lego Mindstorms Ev3-based design activities were applied to Study Group II. A C++ editor-based teaching activity was applied to the control group. The academic achievement test was developed within the scope of this study ( $Kr-20= 0.71$ ). The Logical-Mathematical Intelligence Sub-Scale was used (Cronbach alpha is 0,925). Problem-Solving Inventory was used (Cronbach alpha is 0,86). Data was evaluated using frequency, percentage, arithmetic mean, standard deviation, and Anova analyses. The educational program based on Scratch-related game activities made a more positive contribution to the logical-mathematical thinking skills of students than the educational programs based on Lego Mindstorms Ev3 design activities and traditional teaching activities, and this contribution was especially significant for the mathematical relationship factor.

**Keywords:** *Programming, Games, Learning environment, Problem solving,*

## INTRODUCTION

All tasks from the development of a computer programming problem to the resolution of the problem in a computer environment comprise both thinking and problem-solving processes. For this reason, the concept of computational thinking has been increasingly used in recent times to describe the thought processes required for acquiring and implementing programming skills. According to ISTE (2015), computational thinking not only helps uncover the possible ways to solve a problem, but also uses human creativity and critical thinking to enable computers to enhance human problem-solving capacity. To be successful in computer programming, individuals must first develop and acquire advanced thinking skills, such as problem-solving, logical and mathematical thinking, critical thinking and creative thinking (Fang, 2012; Korkmaz, 2012; Lau & Yuen, 2009; Wang, Geng, Jiang & Liu, 2012). Furthermore, it can also be said that learning computer programming is also an effective way to develop these skills. In this context, it would be accurate to say that problem-solving and logical-mathematical thinking skills are essential for programming education. To ensure that students perform well in computer programming, they first need to develop their problem-solving and logical-mathematical thinking skills, which can be achieved by studying and learning programming. According to ISTE (2015), the information age necessitates that people to learn how, where

and when they need to use digital tools in order to solve problems. ISTE emphasizes that the objective of computational thinking is not just to make students proficient in computer sciences, but also to allow students to apply their computational thinking skills in other classes. Logic is the most important skill humans employ for solving problems, and further developing this skill with the aid of computers and other digital tools has become an important aspect of our daily life and work (Barr et al., 2011).

Since programming is a complex and abstract process, it might be associated with difficulties in understanding the relationship between abstract elements and the outcomes and behaviors they produce, or engender the perception that programming is a set of concepts and processes which, from beginning to end, have no basis in real life. For this reason, different environments are being used in recent years within the framework of programming education. These include programming environments, such as Scratch, Logo and Smalltalk, as well as tangible environments, such as Lego. There are numerous studies in the literature, describing the positive contributions that using the Scratch and Lego programs in programming education have on both academic achievement and other psychometric characteristics (e.g. attitude, etc.) (Calder 2010; Çağiltay Ercil & Fal, 2013; Genç & Karakuş, 2011; Kaučič & Asič, 2011; Varney, Janoudi, Aslam and Graham, 2012; Zaharija, Mladenovic and Boljat, 2013). However, there are only a limited number of studies on the effectiveness of the new methods, approaches and strategies being used to solve the often-emphasized problems encountered in programming education. While there are a number of studies on the use of Lego sets and Scratch type software, we did not identify any studies evaluating or comparing these two environments with respect to their effects on problem-solving, logical-mathematical thinking and academic achievement. In this context, the objective of this study was to determine the effect of the Scratch and Lego Mindstorms Ev3 programming activities on students' academic achievement with respect to computer programming, and on their problem-solving and logical-mathematical thinking skills. A summary of the literature on problem-solving, logical-mathematical thinking, Scratch and Lego Mindstorms Ev3 is presented below.

## THEORETICAL FRAMEWORK

### ***Problem-Solving***

While problem-solving is a concept that has been used for a very long time, its use in practical settings was mainly defined by John Dewey (Bransford and Stein 1984). The problem-solving process involves the following steps: becoming aware of the problem; collecting the necessary information; going to the root cause of the problem; investigating and identifying possible solutions; determining a suitable solution and course; and solving the relevant problem (Mayer, 1998). According to Kalaycı (2001), problem-solving is an essential skill for individuals to adapt to life in society and changes, and to also become independent and successful. According to Gagné (1985), problem-solving is one of the most complex mental skills. As a process, problem-solving involves a broad variety of activities, ranging from trial-and-error to identifying cause-and-effect relationships. A person who solves a problem not only makes use of past learning, but also learns something new in the process. Developing the thinking and problem-solving skills of students is considered as one of the primary objectives of education, and it is believed that enhancing problem-solving skills will further encourage students to perform research and to share what they have learned with one another. More importantly, problem-solving skills provide students the opportunity to assume responsibility for their own learning (Açıkgoz, 2006). Problem-solving is a crucially importance process for all individuals. In his study, McGehee (2001) determined that students achieved better learning when they solved problems by utilizing different disciplines, and that problem-solving consequently had the effect of enhancing their advanced thinking skills as well as their ability to handle problems. A study performed by Arslan (2001) determined that problem-solving skills among teacher candidates varied considerably according to numerous variables. It has also been observed that students with higher problem-solving skills also exhibited higher levels of self-confidence. A study performed by Koray (2003) also demonstrated that employing a creative thinking-based approach for teaching science education to science teacher candidates, had the effect of improving their problem-solving skills. In his study, Yaman (2003) also described that problem-based teaching approaches improved the research and problem-solving skills of classroom teacher candidates.

### ***Logical – Mathematical Thinking***

The Theory of Multiple Intelligence developed by Gardner is one of the fundamental theories that consider intelligence as a primary variable that affects learning. Gardner (1999) defines intelligence as the capacity of an individual to produce a product that has value in one or several cultures; the ability to produce effective solutions to real-world problems; and the ability to discover new or complex problems that require solutions. To provide a broader perspective on the concept of intelligence, Gardner (1997) described 'areas of intelligence' in his theory, which refers to the different abilities and potentials that people possess. According to his theory, there is an important relationship between the type of intelligence and learning styles (Gardner, 1989). The learning styles of individuals tend to differ according to their predominant intelligence characteristics (Harvey, et. al., 1997; Stephen, 2004). An evaluation of programming skills indicates that logical-mathematical intelligence in particular is somewhat more important than other areas (or types of intelligence) for this skill. In fact, Saban (2002) described that individuals with strong logical-mathematical intelligence, are more successful in separating objectives in different categories; establishing logical relationships between events; and digitizing and calculating certain quantitative characteristics of objects in order to establish tangible relationships between events. These skills largely correspond to those required during programming processes (Korkmaz, 2012). For this reason, in this study, we focused solely on this area of the theory of multiple intelligence.

### ***Scratch***

Scratch was developed within the scope of a project conducted by the Massachusetts Institute of Technology (MIT). The aim of this project was to increase the technology-using skills of children in developing societies who are not attending school, and to thereby enable them to produce more meaningful and valuable items through the use of technology (Resnick et al., 2003; Yorulmaz, 2008). Resnick et al. (2003) and Kaučič and Asič (2011) described that Scratch allows computer programming to be understood more easily by everyone. This is made possible by Scratch's basic design features. As a program, Scratch also allows individuals, who are beginning programming education, to understand and acquire programming logic and algorithm thinking skills more easily (Çağiltay Ercil, Fal, 2013; Maloney, Resnick, Rusk, Silverman, Eastmond, 2010). There are numerous studies in the literature regarding Scratch. In a study performed by Genç and Karakuş (2011), students using Scratch described that the program is easy and straightforward to use, and that they even enjoyed using Scratch. A study performed by Calder (2010) described that Scratch contributed to enhancing mathematical thinking, problem-solving, logic and analytical thinking skills, and hence recommended that programming education is provided to students starting from an early age by using the Scratch program. A study performed by Kaučič and Asič, (2011) also described that using Scratch contributes to the development of problem-solving and algorithm skills in children. There are other studies in the literature similarly demonstrating that Scratch has a positive effect on children's programming and basic thinking skills (Genç & Karakuş, 2011; Çağiltay Ercil & Fal, 2013; Kaučič & Asič, 2011).

### ***Lego Mindstorms Ev3***

Lego robotic systems comprise a programming language, a microprocessor, a smart brick, and Lego pieces, such as gears and cogwheels. In addition, the robotic system also uses Logo – a programming language that can operate in coordination with the Lego pieces. Sensors, engines, Lego pieces and programmable bricks constitute the basis of Lego Mindstorms (Güntürkün, 2009). Over the years, the symbol-based visual programming language of Logo has been modified to create RoboLab. Owing to its use of flow diagrams instead of text, it has turned programming into a easily understandable and entertaining activity, eliminating the perception that programming is a boring and tiring activity, associated with low interactivity, teaching methods that are unsuitable for students, and low motivation issues. The robots that are connected to the parts of the set are operated by means of programming with the Robotic Invention System (RIS) – a visual software language – or RoboLab software, and by loading data onto the brick known as RCX with the aid of infrared or a USB cable. Lego sets have many advantages, such as providing real-life experiences to students; enabling both individual and team work; allowing students to actively take part in learning processes; using interdisciplinary knowledge; and presenting alternative ways for solving problems (Çayır, 2010). There are various studies in the literature demonstrating that robotic applications facilitate students' learning of concepts taught in science, technology and engineering classes (Moore, 1999; Papert, 1980). In

addition, there are numerous studies illustrating that robotic applications enhance students' problem-solving skills, multiple-dimensional thinking skills and logical thinking skills, while also increasing their motivation (Varney, Janoudi, Aslam and Graham, 2012 ; Zaharija, Mladenovic and Boljat, 2013).

### **Study Problem**

The aim of this study was to investigate and compare the effect of the Scratch and Lego Mindstorms Ev3 programming activities on academic achievement with respect to computer programming, and on the problem-solving and logical-mathematical thinking skills of students. In this context, the study sought to answer the following questions:

In general, what is the level of logical-mathematical thinking and problem-solving skills of students?

Are the academic achievement, problem-solving skills, and logical-mathematical thinking skills of the students the same as they were before the programming activities were applied?

Have the Lego Mindstorms Ev3- and Scratch-based programming activities contributed more than traditional methods to academic achievement with regards to programming, the problem-solving skills and the logical-mathematical thinking skills of the students?

## **METHODS**

### **Study Design**

This study was a semi-experimental, pretest-posttest study with control groups. The experimental model used in this study is summarized in the table 1 below:

**Table 1. Study Design**

<b>Groups</b>	<b>Pre-test</b>	<b>Experimental Manipulation</b>	<b>Post-test</b>
1.Experimental Group (Scratch)	Academic Achievement Test, Logical-Mathematical Skills Scale, Problem-Solving Skills Inventory	Scratch-Based Game Activities	Academic Achievement Test, Logical-Mathematical Skills Scale, Problem-Solving Skills Inventory
2.Experimental Group (Lego)		Lego Mindstorms Ev3-Based Design Activities	
Control Group		Traditional Method (Application with C++ editor)	

### **Study Groups**

A total of 75 first-year students, including 20 women and 55 men, attending the Electric-Electronic Engineering Department of a university in Turkey and taking a Computer Programming worth three credits were included in the study groups. However, seven students who did not take part in either the pretest or posttest were not included in the study analyses. As such, the number of participants in the study groups decreased to 68. The study groups were formed without any selection criteria – this had the effect of preserving the inherent structure and distribution of the class/year. The distribution of the students between the study and control groups was determined randomly. The distribution with respect to gender of the students constituting the study groups is summarized in Table 2.

**Table 2. Gender Distribution in Study Groups**

	<b>Female</b>	<b>Male</b>	<b>Total</b>
<b>1. Experimental Group (Scratch)</b>	8	12	20
<b>2. Experimental Group (Lego)</b>	6	18	24
<b>Control Group</b>	4	20	24
<b>Total</b>	18	50	68

### **Experimental Operations**

#### *Study Group I (Scratch)*

Scratch-based game activities were applied to Study Group I. In this context, the Scratch environment was first introduced to the students for a period of two weeks, and all programming elements associated with this environment were described to them by providing examples. Until the last week of the application, the students were asked to design a game consisting of at least two characters, two interactions and at least two scenes, using Scratch. Following these sessions on Scratch, subjects relating to the C++ variables, the basic structure of C++, program controls, cycles, conditions, functions and basic ready functions were covered by the students, and examples for each of these subjects were provided using both Scratch and the C++ editor. In addition, every week the students were provided with feedback, as well as recommendations for possible corrections regarding the games they were designing on the Scratch environment. At the end of the six-week period, a gaming contest was organized, based on the tasks assigned to the students at the beginning of the activity; these games were then assessed by both the students and the researcher with respect to the programming processes they included and the quality of the final game product. The first three students received small awards. An example of an image from a game developed by the students is shown in Figure 1.



**Figure 1.** Image of a Sample Game

#### *Study Group II (Lego)*

Lego Mindstorms Ev3-based design activities were applied to Study Group II. In this context, the Lego Mindstorms Ev3 environment was first introduced to the students for a period of two weeks, and all programming elements associated with this environment were described to them by providing examples. Following this, the two projects described in the manuals accompanying the Lego Mindstorms Ev3 set were designed and programmed together with the students. Until the last week of the application, the students were asked to use the Lego Mindstorms Ev3 set to create an entirely novel design. Since it was not possible to provide a Lego Mindstorms Ev3 set to each individual student to create these designs, these students worked in groups. There were eight robots in the activity, and the students were asked to create their own groups on a voluntary basis. In this context, four of the groups had three students, while the four other groups comprised four students. While there was a total of 28 students in these eight groups, four of the students who did not take part in either the pretest or posttest were not included in the study assessments. Following these activities with Lego Mindstorms Ev3, subjects relating to the C++ variables, the basic structure of C++, program controls, cycles, conditions, functions and basic ready functions, were covered with the students and examples for each one of these subjects were provided through both Lego Mindstorms Ev3 and the C++ editor. In addition, the students were provided with feedback every week, as well as recommendations for possible corrections regarding their Lego Mindstorms Ev3 projects. At the end of the six-week period, a project-related contest was organized, based on the tasks that were assigned to the students at the beginning of the activity; both the students and the researcher then assessed these games with regard to the programming processes they included, and the quality of the final product. An example of an image from a project developed by the students is shown in Figure 2.



**Figure 2.** Project Activities

#### **Control Group**

A C++ editor-based teaching activity was applied to the control group. In this context, the C++ environment was introduced to the students for a period of two weeks, and all programming elements associated with this environment, as well as concepts such as compiler and debuggers, were described to the students with the use of examples. At the last week of the activity, students were requested to design a program using the C++ editor to solve a pre-defined problem. After this two-week period, subjects relating to the C++ variables, the basic structure of C++, program controls, cycles, conditions, functions and basic ready functions were covered with the students, and examples for each one of these subjects were provided through the C++ editor. In addition, the students were provided with feedback and recommendations regarding their projects on a weekly basis. At the end of the six-week period, a contest was organized, based on the project tasks that were assigned at the beginning of the activity; these projects were then assessed by both the students and the researcher with respect to the programming processes they included and the quality of the final product.

The experimental procedures applied in both the study and the control groups are summarized in Table 3.

**Table 3. Experimental procedures applied to the Study and Control Groups**

Week	Subject	Study Group I	Study Group II	Control Group
1	Administration of pretest, and detailed briefing on the activity	X	X	X
	Lego Applications		X	
2-3	General structure of Scratch	X		
	General structure of the C++ Editor			X
4	General structure of C; Operators; Priority of operators; Data types;	X	X	X
5	Assignment and Input-Output commands; Printf; Scanf	X	X	X
6	Selection (flow) commands; If-Else/Switch	X	X	X
7	Cycle commands; While/for/do while/ break/ continue	X	X	X
8	Functions/Indicator	X	X	X
9	Series/Strings	X	X	X
	Lego Project contest		X	
	Scratch-based game contest	X		
10	Project contest with C++ editor			X
	Administration of posttest.	X	X	X

As shown in Table 3, the activity schedule applied to each group differed in the first two weeks (with regards to the programming environment introduced to the students), and the theme of the contests performed at the end of the activity. This ensured that the only experimental manipulation in the study was with respect to the application of the Scratch and Lego environments.

### ***Data Collection Tools***

#### *Academic achievement Test:*

The academic achievement test was developed within the scope of this study. The test consisted of 30 multiple-choice items with five possible answers on the variables of C++, the basic structure of C++, program controls, cycles, conditions, functions and basic ready functions. Following a pilot application performed on 148 students who previously took the C++ course, five items with a differentiation level below 0.3, and which were unlikely to affect the scope validity of the test, were removed from the scale. The final scale consisted of 25 items, whose differentiation coefficients varied between 0.296 and 0.740, and whose difficulty index was 0.51. The Kr-20 internal consistency coefficient of the scale was determined as 0.71.

#### *The Logical and Mathematical Skill Scale*

The Logical-Mathematical Intelligence Sub-Scale developed by Yeşil and Korkmaz (2010) was used to determine the logical-mathematical skill levels of the students. While this scale is actually a sub-scale of the Multiple Intelligence Scale, Yeşil and Korkmaz (2010) described that the sub-scales can be used separately. The scale consisted of a total of 21 items organized under three factors.

**Table 4. Structure Validity and Internal Consistency**

Factors	Items	KMO	Bartlett	Eigenvalue	Variances	Cronbach Alpha
Mathematical Transfer	8					0.930
Mathematical Relationship	7	0.952	p<0.001	8.742	41.62	0.885
Logical Relationship	6			2.932	13. 69	0.769
Total	21			-	47.34	0.925

The logical-mathematical intelligence apperception sub-scale was divided into two different factors, while the first of these two factors was further divided into two sub-factors. Total variance for these two factors was 55.591%. The first sub-factor of the first factor (F1A) has a load varying between 0.614 and 0.794, while the load of the second sub-factor (F2A) varied between 0.517 and 0.796. The first factor (F1) accounted for 41.627% of the total variance. The second factor had a load varying between 0.518 and 0.769, and accounted for 13.964% of the total variance.

#### *The Problem-Solving Inventory*

We also used the Problem-Solving Inventory, which was developed in 1982 by Heppner and Peterson, to assess the problem-solving skills of education faculty students, and whose reliability study in Turkey was conducted by Taylan in 1990. This inventory is an assessment tool that evaluates the opinions and approaches of individuals about their own problem-solving behavior (which determines the stages of problem solving). The inventory is a six-point Likert type scale consisting of 35 items. The internal consistency coefficient of the scale is 0.86 (Taylan, 1990: as cited by Öztürk Can, Öner, Çelebi, 2009). The scale has 17 negative and 18 positive items. Items number 1, 2, 3, 4, 9, 11, 13, 14, 17, 21, 22, 25, 26, 29, 30, 32 and 34 are negative, and were coded inversely when being analyzed within the scope of this study. The original scale developed by Heppner and Peterson (1982) consists of three factors, which are Belief in Problem-Solving (items 5, 10, 11, 12, 19, 23, 24, 27, 33, 34 and 35), Avoidance (items 1, 2, 4, 6, 7, 8, 13, 15, 16, 17, 18, 20, 21, 28, 30 and 31) and Personal Control (Items 3, 14, 25, 26 and 32).

### ***Data Analysis***

The scores obtained, based on the responses to the 5- and 6-point Likert type data collection tools used during the study, did not exhibit a standard characteristic due to the differences in the number of items

of the sub-factors. For this reason, it was deemed suitable to convert the raw scores obtained from the scales into a standard scoring ranging from 20 (at the lowest) to 100 (at the highest). In this context, data were evaluated using frequency, percentage, arithmetic mean, standard deviation, and ANOVA analyses.

## RESULTS

### ***Problem-Solving and Logical-Mathematical Skill Levels of the Students***

Data regarding the students' problem-solving and logical-mathematical skill levels are summarized in Table 5.

**Table 5. The Problem-Solving and Logical-Mathematical Thinking Skills of the Students**

Variables		N	Min	Ma x	Mea n	Std. Dev	Skew	Kur.
Prob. Solv.	Problem-solving beliefs	68	33	96	72.0	13.2	-	0.258
	Avoidance		37	92	67.5	10.6	-	0.312
	Self-Control		27	97	64.9	15.6	-	-
	Mathematical Transfer		30	100	69.7	18.3	-	-
	Mathematical Relationship		26	97	63.1	14.6	-	-
	Logical Relationship		33	100	78.7	14.8	-	0.673

As shown in Table 5, the pretest results indicated that, with regards to the students' problem-solving skills, the scores for the belief in problem-solving factor varied between 33 and 96, and had a mean value of 72.04. The scores for the avoidance factor varied between 37 and 92, and had a mean value of 67.53. The scores for the personal control factor varied between 27 and 97, and had a mean value of 64.93. These results indicated that the students had a medium-level problem-solving skill, with the highest mean score being observed for the belief in problem-solving factor, while the lowest mean score was observed for the personal control factor. On the other hand, based on the Skewness and Kurtosis values and the Shapiro-Wilk values (Belief in Problem-Solving: Shapiro-Wilk<sub>68</sub>=0.966, p>0.05; Avoidance: Shapiro-Wilk<sub>68</sub>=0.965, p>0.05; Personal Control: Shapiro-Wilk<sub>68</sub>=0.970, p>0.05), it is possible to say that the student scores exhibited a normal distribution (Tabachnick & Fidell, 2013).

An evaluation of the logical-mathematical skill levels on Table 5 indicates that the students' scores for the mathematical transfer factor varied between 30 and 100, and that their mean score was 69.72. Furthermore, the scores for the mathematical relationship factor varied between 26 and 97, and had a mean value of 63.13, while the scores for the logical factor varied between 33 and 100, and had a mean value of 78.75. According to these results, it is possible to say that the students exhibited a medium-level mathematical-logical thinking skill, and that the highest mean value was observed with the logical relationship factor, while the lowest mean value was observed with the mathematical relationship factor. On the other hand, based on the Skewness and Kurtosis values and the Shapiro-Wilk values (Mathematical transfer: Shapiro-Wilk<sub>68</sub>=0.969, p>0.05; Mathematical relationship: Shapiro-Wilk<sub>68</sub>=0.988 p>0.05; Logical relationship: Shapiro-Wilk<sub>68</sub>=0.975, p>0.05), it is possible to say that the student scores exhibited a normal distribution (Tabachnick & Fidell, 2013).

### ***Results Regarding the Match between the Groups Before the Application of the Study Procedures***

The data regarding the pretest averages for problem-solving, logical-mathematical thinking and academic achievement before the application of the study procedures are summarized in Table 6.

**Table 6. Results Regarding the Pretest Score Averages**

Variables and Groups		N	Mean	Std. Dev.
Problem-solving beliefs	I. Exp. G. (Scratch)	20	69.25	10.69
	II. Exp. G. (Lego)	24	68.83	15.62
	Control Group	24	77.58	11.25
Avoidance	I. Exp. G. (Scratch)	20	65.10	11.76
	II. Exp. G. (Lego)	24	68.92	8.89
	Control Group	24	68.17	11.47
Self-Control	I. Exp. G. (Scratch)	20	62.30	14.99
	II. Exp. G. (Lego)	24	61.42	14.85
	Control Group	24	70.63	15.87
Mathematical Transfer	I. Exp. G. (Scratch)	20	73.65	17.51
	II. Exp. G. (Lego)	24	70.63	18.59
	Control Group	24	65.54	18.56
Mathematical Relationship	I. Exp. G. (Scratch)	20	67.55	13.37
	II. Exp. G. (Lego)	24	65.63	14.25
	Control Group	24	56.96	14.58
Logical Relationship	I. Exp. G. (Scratch)	20	79.40	11.60
	II. Exp. G. (Lego)	24	78.041	16.99
	Control Group	24	78.92	15.54
Academic Achievement	I. Exp. G. (Scratch)	20	26.00	10.82
	II. Exp. G. (Lego)	24	25.33	13.12
	Control Group	24	28.67	11.54

An evaluation of the different groups' pretest scores on Table 6 indicates that the mean problem-solving skills and academic achievement scores of the control group were slightly higher than that of the study groups. With respect to the logical-mathematical thinking skills, it was determined that Study Group I had slightly higher scores. The analysis of variance results on whether these differences were significant are summarized in Table 7.

**Table 7. Results Regarding the Match in the Pretest Mean Scores of the Different Groups**

Variables		Sum of Squares	df	Mean Square	F	Sig.	LSD
Problem-solving beliefs	Between Groups	1139.951	2	569.975	3.464	0.037	Cont.
	Within Groups	10694.917	65	164.537			and others
	Total	11834.868	67				
Avoidance	Between Groups	173.975	2	86.987	0.757	0.473	
	Within Groups	7464.967	65	114.846			-
	Total	7638.941	67				
Self-Control	Between Groups	1212.974	2	606.487	2.604	0.082	
	Within Groups	15141.658	65	232.949			-
	Total	16354.632	67				
Mathematical Transfer	Between Groups	747.558	2	373.779	1.120	0.332	
	Within Groups	21690.133	65	333.694			-
	Total	22437.691	67				
Mathematical Relationship	Between Groups	1454.275	2	727.138	3.648	0.032	I. Exp and others
	Within Groups	12957.533	65	199.347			
	Total	14411.809	67				
Logical Relationship	Between Groups	21.158	2	10.579	0.047	0.955	
	Within Groups	14757.592	65	227.040			-
	Total	14778.750	67				

Variables		Sum of Squares	df	Mean Square	F	Sig.	LSD
Academic Achievement	Between Groups	147.451	2	73.725	0.518	0.598	-
	Within Groups	9242.667	65	142.195			
	Total	9390.118	67				

An evaluation of Table 7 reveals that there was a significant difference between the pretest scores of the groups with respect to the belief in problem-solving ( $F_{(2-65)}=3.464$ .  $p<0.05$ ) and the mathematical relationship ( $F_{(2-65)}=3.648$ .  $p<0.05$ ) factors. An evaluation of the LSD results shows that significant differences existed between the control group and the other groups with regards to the belief in problem-solving factor, and between Study Group I and the other groups with regards to the mathematical transfer factor. While the differences in the other factors were not statistically significant, there was still a certain quantitative difference. To control the variations between the groups, the score differences listed below were used for the pretest and posttest scores. Since score differences were used, there was no need to perform a covariance analysis within the scope of this study.

#### ***The Effect of the Scratch and LEGO Mindstorms Ev3 Design Activities***

The mean pretest and posttest scores for problem-solving, logical-mathematical thinking and academic achievement following the application of the Scratch and Lego Mindstorms Ev3 design activities and the C++ editor-based traditional methods are summarized in Table 8.

**Table 8. Results Regarding Pretest and Posttest Mean Scores**

Variables and Groups		N	Mean	Std. Dev.
Problem-solving beliefs	I. Exp. G. (Scracth)	20	2.45	9.79
	II. Exp. G. (Lego)	24	8.88	17.96
	Control Group	24	.63	13.40
Avoidance	I. Exp. G. (Scracth)	20	1.30	9.61
	II. Exp. G. (Lego)	24	1.75	9.74
	Control Group	24	0.30	9.36
Self-Control	I. Exp. G. (Scracth)	20	6.17	16.43
	II. Exp. G. (Lego)	24	7.17	17.16
	Control Group	24	5.15	9.93
Mathematical Transfer	I. Exp. G. (Scracth)	20	7.46	16.09
	II. Exp. G. (Lego)	24	4.50	22.74
	Control Group	24	1.25	19.86
Mathematical Relationship	I. Exp. G. (Scracth)	20	14.58	13.10
	II. Exp. G. (Lego)	24	5.38	17.27
	Control Group	24	3.75	8.87
Logical Relationship	I. Exp. G. (Scracth)	20	10.71	13.64
	II. Exp. G. (Lego)	24	8.04	20.18
	Control Group	24	2.45	11.43
Academic Achievement	I. Exp. G. (Scracth)	20	29.60	13.26
	II. Exp. G. (Lego)	24	44.33	15.73
	Control Group	24	20.00	11.07

An evaluation of the pretest and posttest score differences in Table 8 shows that the highest mean scores for the problem-solving and personal control factors were in Study Group II, while the lowest mean scores were in the control group. The highest mean score for the avoidance factor was observed in Study Group I. With regards to logical-mathematical thinking, the highest mean scores were observed in all factors in Study Group I, while the lowest mean scores were observed in the control group. With regards to academic achievement, the highest mean scores belonged to Study Group II, while the lowest mean score belonged to the control group. The results of the analysis of variance on whether the differences between the mean score modifiers were significant are summarized in Table 9.

**Table 9. Results Regarding the Differentiation between the Pretest and Posttest Score Differences**

Variables		Sum of Squares	df	Mean Square	F	Sig.	LSD
Problem-solving beliefs	Between Groups	1122.609	2	561.304	2.728	.043	II. Exp and others
	Within Groups	13375.200	65	205.772			
	Total	14497.809	67				
Avoidance	Between Groups	49.327	2	24.663	.269	.765	-
	Within Groups	5951.658	65	91.564			
	Total	6000.985	67				
Self-Control	Between Groups	44.475	2	22.237	.097	.907	-
	Within Groups	14851.217	65	228.480			
	Total	14895.691	67				
Mathematical Transfer	Between Groups	420.762	2	210.381	.540	.586	-
	Within Groups	25341.708	65	389.872			
	Total	25762.471	67				
Mathematical Relationship	Between Groups	1565.321	2	782.661	3.954	.024	I. Exp and others
	Within Groups	12865.208	65	197.926			
	Total	14430.529	67				
Logical Relationship	Between Groups	762.354	2	381.177	1.536	.223	-
	Within Groups	16128.867	65	248.136			
	Total	16891.221	67				
Academic Achievement	Between Groups	7198.337	2	3599.169	19.742	.000	With Cont. G and Other G
	Within Groups	11850.133	65	182.310			
	Total	19048.471	67				

An evaluation of Table 9 shows a significant difference between the groups with respect to the change between their pretest and posttest scores for the belief in problem-solving factor ( $F_{(2-65)}=2.728$ .  $p<0.05$ ). An evaluation of the LSD results indicated that the difference for the belief in problem-solving factor was mainly between the Study Group II and the Study Group I. An evaluation of the mean values on Table 7 shows that the mean score difference between pretest and posttest for Study Group II was higher than the score differences observed in Study Group I (Mean=2.45) and the control group (Mean=0.63). No significant differences were observed in the other two factors of problem-solving; however, an evaluation of the mean values for the different groups still indicated that, in both factors, Study Group II had higher mean values. In this context, it is possible to say that an educational program based on Lego Mindstorms Ev3 design activities had a more positive contribution to the problem-solving skills of students than educational programs based on Scratch-related game activities and traditional teaching activities, and that this contribution was particularly significant for the belief in problem-solving factor.

An evaluation of the results for the logical-mathematical thinking skills revealed that, for the mathematical relationship factor, the difference between the pretest and posttest scores of the groups was significant ( $F_{(2-65)}=3.954$ .  $p<0.05$ ). The LSD results showed that, with regards to the mathematical relationship factor, Study Group I differed from Study Group II and the control group. The mean values in Table 7 show that the mean difference between the pretest and posttest scores for Study Group I (Mean=14.58) was greater than that for both Study Group II (Mean=5.38) and the control group (Mean=3.75). No significant differences were observed in the other two factors of logical-mathematical thinking; however, in both factors, Study Group I had higher mean values, even though the differences were not significant. In this context, it is possible that an educational program based on Scratch-related game activities had a more positive contribution to the logical-mathematical thinking skills of students than educational programs based on Lego Mindstorms Ev3 design activities and traditional teaching activities, and that this contribution was especially significant for the mathematical relationship factor.

An evaluation of the academic achievement scores indicates that the difference between the pretest and posttest scores was significant in the different groups ( $F_{(2-65)}=19.7422$ .  $p<0.05$ ). The LSD results showed that the level of pretest-posttest difference in the academic achievement score for Study Group II differed

from Study Group I and control group. The mean values in Table 7 show that the mean difference between the pretest and posttest scores for Study Group II (Mean=44.33) was greater than that for both Study Group I (Mean=29.60) and the control group (Mean=20.00). In this context, it is possible to say that an educational program based on Lego Mindstorms Ev3 design activities had a more positive contribution to the academic achievement of students than educational programs based on Scratch-related game activities and traditional teaching activities. In addition, while there were no significant differences, Scratch-based game activities were still nevertheless found to make a greater contribution to the academic achievement of students than traditional teaching methods.

## CONCLUSIONS AND RECOMMENDATIONS

The students had medium-level mean scores in problem-solving, with the highest mean scores being observed in the belief in problem-solving, and the lowest scores being observed in the personal control factor. The students similarly had medium-level mean scores in logical-mathematical thinking skills, with the highest mean scores being observed in the logical relationship factor, and the lowest scores being observed in the mathematical relationship factor.

The educational program based on Lego Mindstorms Ev3 design activities had a more positive contribution to the problem-solving skills of students than the educational programs based on Scratch-related game activities and traditional teaching activities, and this contribution was particularly significant for the belief in problem-solving factor. This observation is in agreement with the existing literature. In addition, there are numerous studies in the literature demonstrating that Lego increases problem-solving skills (Gibbon, 2007; Lenamond, 1991; McDaniel, 2004). Teaching activities with Lego are generally project-based teaching applications that require students to solve a pre-determined problem by developing their own approaches; as such, it is expected that Lego-based teaching activities will have a positive impact of problem-solving skills. On the other hand, it was also observed that educational applications involving Lego Mindstorms Ev3 design activities make a greater positive contribution to students' academic achievements with respect to C++ programming than Scratch-based game activities. This observation is also in agreement with the literature. For example, a study by Özdogru (2013) described that students achieved better problem-solving skills through Lego-related applications. Another study conducted by Koumoulos (2013), described that students participating in a robotics course had better academic achievement than students who did not.

The educational program based on Scratch-related game activities made a more positive contribution to the logical-mathematical thinking skills of students than the educational programs based on Lego Mindstorms Ev3 design activities and traditional teaching activities, and this contribution was especially significant for the mathematical relationship factor. In addition, while not being significant, it was still observed that educational applications using Scratch-based game activities contributed more to the academic achievement of the students in this study than did traditional educational approaches. This observation is in agreement with the existing literature. There are numerous studies in the literature describing that Scratch contributes to different psychometric characteristics of students. For example, in a study performed by Genç and Karakuş (2011), it was described that students generally had positive views about Scratch, that Scratch allowed lasting learning through design activities, and that the students embraced the blog-supported learning method. A study performed by Kobsiripat (2015) to determine the effect of programming activities with Scratch on creativity levels determined that this environment is very suitable for learning, and that it helped develop the creativity of students. In addition to this study, studies performed by Kordaki (2012), [Ferrer-Mico](#), [Prats-Fernández](#), [Redo-Sánchezb](#) (2012) and [Garcia Quan](#) (2015) have also described similar findings.

There are only a limited number of studies in the literature investigating the effectiveness of the new methods, approaches and strategies for solving the difficulties encountered during programming education. While there are a number of studies regarding the use of Lego sets and Scratch type software in different levels of programming education, there are only a few studies on their use in engineering courses and by undergraduate students. Furthermore, there are no previous studies that compare these two environments with respect to their effect and contribution on problem-solving, logical-mathematical thinking and academic achievement. In this context, the results of this study, determining the contributions of Lego Mindstorms Ev3 and Scratch to programming education, showed that Lego Mindstorms Ev3 made a greater contribution to

the students' problem-solving skills, while Scratch contributed more to the students' logical-mathematical thinking skills. In addition, it was also observed that Lego Mindstorms Ev3 contributed to a greater extent than Scratch to the students' academic achievement. However, it was also noted that Scratch more contribution to the student's academic achievement than the traditional teaching method. In light of these results, it is possible to recommend the use of Lego Mindstorms Ev3 applications in programs and courses aiming to develop the algorithmic thinking and programming skills of undergraduate students who have recently begun their programming education. However, because these sets are rather expensive, these courses and programs can instead be supported with Scratch software, in cases where Lego Mindstorms Ev3 sets cannot be acquired.

## ACKNOWLEDGEMENT

This research has been executed within the scope of SEB-BAP 14-37 project endorsed by Amasya University Project Coordination Unit of Scientific Researches.

## REFERENCES

Açıköz, K. Ü. (2006). Aktif Öğrenme [Activ Learning]. İzmir: Biliş Pub.

Arslan, C. (2001). An Analysis of teachers and student teachers problem solving skills through different variables. Master dissertation, Selcuk University, Konya.

Barr, D., Harrison, J. & Conery, L.(2011).Computational Thinking: A Dijital Age Skill for Everyone, Available at: <http://files.eric.ed.gov/fulltext/EJ918910.pdf>

Bransford, J., Stein, B. (1984). The IDEAL problem solver. New York: W. H. Freeman.

Çağiltay Ercil, N., & Fal, M. (2013). Scratch ile Programlamayı Öğreniyorum [I am learning computer programming with Scratch]. Ankara: ODTÜ Pub.

Calder, N. (2010). Using Scratch:An Integrated Problem-solving Approach to Mathematical Thinking. APMC 15 (4), 9-14.

Çayır, E. (2010). Assignment the effect of learning environment supported by lego-logo on science process skill and self concept, Master dissertation, Sakarya University, Sakarya.

Fang, X. (2012). Application of the participatory method to the computer fundamentals course, Affective Computing and Intelligent Interaction. Advances in Intelligent and Soft Computing, 137: 185-189.

*Ferrer-Mico*, T., *Prats-Fernández*, M. À, *Redo-Sanchez*, A. (2012). Impact of Scratch Programming on Students' Understanding of Their Own Learning Process. *Procedia - Social and Behavioral Sciences Volume 46*: 1219–1223. [doi:10.1016/j.sbspro.2012.05.278](https://doi.org/10.1016/j.sbspro.2012.05.278)

Gagne, R. M. (1985). The conditions of learning and theory of instruction. New York: Holt, Rinehart and Winston.

*Garcia Quan*, C. (2015). Student Teachers Evaluating and Assessing SCRATCH in the Applied Linguistics Classroom. *Procedia - Social and Behavioral Sciences. Volume 174*, : 1450–1456. [doi:10.1016/j.sbspro.2015.01.774](https://doi.org/10.1016/j.sbspro.2015.01.774)

Gardner, H. (1989). Educational implications of the theory of multiple intelligences. *Educational Researcher*, Vol. 18, No. 8, 4-10. DOI: 10.3102/0013189X018008004

Gardner, H. (1997). Multiple Intelligences as a partner in school improvement. *Educational Leadership*, 55(1), ss. 20–21.

Gardner, H. (1999) *Intelligence reframed. Multiple intelligences for the 21st century*. New York: Basic Books. 292p.

Genç. Z., Karakuş, S. (2011). Learning Through Design: Using Scratch In Instructional Computer Games Design. 5 th International Computer & Instructional Technologies Symposium, 22-24 September Fırat University, Elazığ, p:981-987.

Gibbon, L.W. (2007). Effects of Lego Mindstorms on Convergent and Divergent Problem Solving and Spatial Abilities in Fifth and Sixth Grade Students. Doctoral dissertation, Seattle Pacific University, Washington, USA.

Güntürkün, E. (2009). Historical and structural development of construction toys (Using the example of LEGO company), Master dissertation, Marmara University, İstanbul.

Harvey, S., Richard, S., Matthew, P. (1997). Integrating learning styles and multiple intelligences. *Educational Leadership*, v55 n1. Eric No: EJ550527

Heppner, P. P., Peterson C. H.(1982). The Development and Implications of a Personal Problem-Solving Inventory. *Journal of Counseling Psychology*. 1982, Vol. 29, No. 1,66-75.

Iste. (2015). CT Leadership toolkit. Available at <http://www.iste.org/docs/ct-documents/ct-leadership-toolkit.pdf?sfvrsn=4>

Kalaycı, N. (2001). *Sosyal Bilgilerde Problem Çözme ve Uygulamalar* .[Problem solving and applications in social science]. Ankara: Gazi pub

Kaučič, B., & Asič, T. (2011). Improving Introductory Programming with Stratch? MIPRO, p:1095-1100.

Kobsiripat, W. (2015). Effects of the Media to Promote the Scratch Programming Capabilities Creativity of Elementary School Students. *Procedia - Social and Behavioral Sciences*. Volume 174, Pages 227–232. [doi:10.1016/j.sbspro.2015.01.651](https://doi.org/10.1016/j.sbspro.2015.01.651)

Koray, Ö. C. (2003). The Influence of science education based on creative thinking on learning products. PhD dissertation, Gazi University, Ankara.

Kordaki. M. (2012). *Diverse Categories of Programming Learning Activities could be Performed within Scratch*. Procedia - Social and Behavioral Sciences, Volume 46: 1162-1166. [doi:10.1016/j.sbspro.2012.05.267](https://doi.org/10.1016/j.sbspro.2012.05.267)

Korkmaz, Ö. (2012). The Impact of Critical Thinking and Logical-Mathematical Intelligence on Algorithmic Design Skills. *Journal of Educational Computing Research*, 46(2):173-193. DOI: 10.2190/EC.46.2.d.

Koumoulos, M.G. (2013). The Academic Differences Between Students Involved in School –Based Robotics Programs and Students Not Involved in School- Based Robotic Programs. Doctoral dissertation, St. John University, New York, USA.

Lau, W.W.F., Yuen, A.H.K. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57: 1202–1213. doi:10.1016/j.compedu.2011.01.002

Lenamond, D.B. (1991). A Comparison of the Effects of Lego Tc Logo and Problem Solving Software on Problem Solving Skills. Master dissertation, University of Houston- Clear Lake, Houston, Texas, USA

Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), 1-16.

Mayer, R. E. (1998). "Cognitive, Metacognitive And Motivational Aspects Of Problem Solving". *Instructional Science*, 26, 49-63.

McDaniel, A.R. (2004). How A Group of Elementary Deaf Students Interact with LEGO LOGO Activities: A Case Study. Doctoral dissertation, The University of Tennessee, Knoxville, USA.

McGehee, J. J. (2001). Developing Interdisciplinary Units: A Strategy Basd on Problem Solving. *School Science and Mathematics*, 101,7.

Moore, V.S. (1999). Robotics: Design through Geometry. *Technology Teacher*, 59(3), 17-22.

Özdoğru, E. (2013). The effect of Lego programme based science and technology education on the students academic achievement, science process skills and their attitudes toward Science and Technology course for physical facts learning field. Master dissertation. Dokuz Eylül University, İzmir.

Öztürk C., H., Öner, Ö. İ., Çelebi, E. (2009). Üniversite Öğrencilerinde Eğitimin Sorun Çözme Becerisine Etkisinin İncelenmesi. [Evaluation of the Effects of Problem Solving Skills Training in University Students]. *Fırat The Journal of Health Services*. Vol 4, Iss: 10.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Resnick, M., Kafai, Y., Maloney, J., Rusk, N., Burd, L., & Silverman, B. (2003). A Networked, Media-Rich Programming Envi-ronment to Enhance Technological Fluency at After-School Centers in Economically Disadvantaged Communities. *Proposal to National Science Foundation*.

Saban, A. (2002). Çoklu zeka teorisi ve e itimi [The theory and teaching of multiple intelligences]. Ankara: Nobel Publishing.

Stephen, D. (2004). Multiple intelligences and learning styles: Two complementary dimensions. *The Teachers College Record*, V. 106, N. 1. Blackwell Publishing.

Tabachnick, B.G., Fidell L.S. (2013). *Using Multivariate Statistics* (sixth ed.). Pearson, Boston

Varnado, T. E. (2005). The Effects of a Technological Problem Solving Activity on FIRST LEGO League Participants' Problem Solving Style and Performance. Doctoral dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Varney, M. W., Janoudi, A., Aslam, D. M. ve Graham, D. (2012). Building Young Engineers: TASEM for Third Graders in Woodcreek Magnet Elementary School. IEEE Transactions on Education, 55(1).

Wang, Y., Li, H., Feng, Y., Jiang, Y., & Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. Computers & Education, 59: 412–422. doi:10.1016/j.compedu.2012.01.007.

Yaman, S. (2003). The Effects on the learning outputs of problem based learning in science education. PhD dissertation. Gazi University, Ankara.

Yeşil, R., Korkmaz, Ö. (2010). Reliability and validity analysis of the multiple intelligence perception scale. Education. Project Innovation Inc. Volume 131, Issue1. 8-32.

Yorulmaz, M. (2008). Internet Kafelerin Daha Faydalı Kullanılabilmeleri için Bir Öneri: Scratch [A proposal for the use of the Internet cafe can more useful: Scractch]. inet-tr'08, Ankara

Zaharija, G., Mladenovića, S., & Boljat, I. (2013). Introducing the Basic Programming Concepts to Elementary School Children. 4 th International Conference on New Horizons in Education, Social and Behavioral Sciences, 106, 1576-1584.